King Saud University

CSC 311 - Design and Analysis of Algorithms



- الد عال

Midterm Exam, Fall 2022

Wednesday Oct 5th, 2022

Exam time: 08:00-9:30 P.M.

Student's name:

Problem 1 (11 p

(a) Give the following functions a number in order of increasing asymptotic growth rate. If two functions have the same asymptotic growth rate, give them the same number.

Г	Function	Rank
100 H 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	$3 \log n + \log \log \log (n^3)$	
	$3 \log n + \log \log \log (n^3)$ $5 \lg n^5$	
	n^3	2
	$\overline{\lg n}$	
		3
	$n^3 + 6 \log n - 5$ $n^4 - 16^{\log n} + 3 n^2$ 4^{3n}	4 //
	4 ³ⁿ	5

(b) Using the definition of θ , find g(n), C_1 , C_2 , and n_0 in the following:

[4 pts]

no=1

$$6n^4 - 3n^3 + n \log n \in \theta(g(n))$$

$$\Omega(n^{4}) = 6n^{4} - 3n^{3} + n \log n > 5n^{4}$$

$$n_{2} = \frac{4}{6-5} - 4$$

$$\Theta(n^{q})$$
 $\begin{cases} c_{1}=10 \\ c_{2}=5 \\ n_{0} \\ m(x(N_{01}, N_{02})=4 \end{cases} = 4$

I liniversity

King Saud University

480 of computer and Information Sciences

C 311 – Design and Analysis of Algorithms

جامعة الملك سعود كلية علوم الحاسب والمعلومات 311 عال _

[4 pts]

(c) Prove that log(n!) is O(n log n).

nl=n-1xn-2 xn-3--- complicity nl so tog is mineralize the complixity to O(nlogn)

Problem 2 (7 points)

Consider the pseudo-code below:

a- Which problem does this algorithm solve?

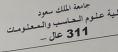
It's sort the arrays by arrange? the even numbers in the first them before any odd number (the Ha)



King Saud University

College of computer and Information Sciences

CSC 311 - Design and Analysis of Algorithms



b- What is the time complexity (*Theta (?)*) of the following algorithm? Prove your answer (explain each

step).

$$T(n) = 3n+3 \qquad \Theta(n)$$

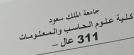
$$O(n) = \frac{1}{3} = \frac{3}{3-2} = 3$$

$$O(n) = \frac{3}{3-2} = 3$$

$$O(n) = \frac{3}{3-2} = 3$$

$$O(n) = \frac{3}{3-2} = 3$$

King Saud University College of computer and Information Sciences CSC 311 - Design and Analysis of Algorithms



Consider the pseudo-code below, and assume the array A[..] is sorted in an increasing order:

```
S(A[..], key, imin, imax)
 if (imax < imin) t
       return KEY_NOT_FOUND; 1
     imid = (imin+imax)/2;
     if (A[imid] > key)
      return S(A, key, imin, imid-1); T(4)
     else if (A[imid] < key) —
                                             て(1)=丁(2)+9
      return S(A, key, imid+1, imax); -
     else —
      return imid; -
```

a- Which problem does this algorithm solve?

Bainty Search tokind the mid edlement

b- What is the design technique used in this solution. Explain your answer.

Devid and concer technique

because me need be depid the element into two arroy eable time and then return the result

علام علوم المعلوما 311 عال -

c- What is the time complexity of the following algorithm? Prove your answer.

$$T(n) = T(\frac{h}{2}) + 4$$

$$T(\frac{n}{2}) = [T(\frac{n}{4}) + 4] + 4$$

$$T(\frac{n}{4}) = T(\frac{n}{16}) + 4 + 4 + 4$$
after $k : T(\frac{n}{2}k) + 4k$

$$(1) + 4 \log n = 1 + 4 \log n$$

$$\frac{n}{2}k = 1$$

$$n = 2k$$

$$k = \log n$$

$$C = 5$$

$$n_0 = 1$$

Problem 4 (7 points)

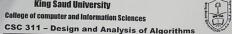
Consider the following recurrence relation:

$$T(n) = 4T(\frac{n}{4}) + 3n.$$

Solve this recurrence relation using recursive substitutions. Find g(n), where T(n) = O(g(n)).

next page

King Saud University



$$T(n) = 9 T(n) + 3 n$$

$$T(\frac{n}{q}) = 9 \left[9 T(\frac{n}{16}) + 3 \frac{n}{q} \right] + 3 n = \frac{16}{16} T(\frac{n}{16}) + 12 \frac{n}{q} + 3 n$$

$$T(\frac{n}{16}) = \frac{69}{16} T(\frac{n}{69}) + 69 \frac{n}{16} + 12 \frac{n}{q} + 3 n$$

$$= \frac{69}{16} T(\frac{n}{69}) + 69 \frac{n}{16} + 12 \frac{n}{4} + 3 n$$

$$after k : 9 T(\frac{n}{9}) + 3 n = \frac{1}{16} T(\frac{n}{16}) + 12 \frac{n}{4} + 3 n$$

$$after k : 9 T(\frac{n}{9}) + 3 n = \frac{1}{16} T(\frac{n}{16}) + 12 \frac{n}{9} + 3 n$$

$$\frac{n}{9} = 1 = 9 \cdot 9 \cdot 9 \cdot 7 \cdot 1 + 3 \cdot 1$$

Solve the following recurrence using the Master theorem by giving tight θ -notation bounds. Justify your answers.

(a)
$$T(n) = 3T(n/3) + 3 \log^3(n)$$
 $a = 3$ $b = 3$ $f(n) = 3 \log^3(n)$

$$f(n) = 3 \log^3(n)$$

$$3\log^3(n) \ll n^{\log 3} = n$$



Problem 6 (10 points)

There are n parking spots numbered from 1 to n and you are told that there are k cars in the first k spots (one car in each spot), and all a! in each spot), and all other spots are empty. How to find the value of k?

(a) Suggest TWO algorithm design techniques and give a high-level description of the TWO algorithms to solve the problem (find k)?

Deide and concert

find
$$k(A[1.-n])$$

mid $= \frac{n}{2}$

If $(mid empty)$

find $k(A[1.-mid-1])$

return mid

$$T(n) = T(\frac{n}{2}) + 3$$

(b) Analyze the complexity of your TWO algorithms?

$$T(n) = 3n+2$$

$$O(n) = 5$$

$$n_0 = 1$$

$$T(n) = T(\frac{n}{2}) + 3$$

$$T(n) \le 1 + 3 \log n$$

$$O(\log n)$$

$$C = 4$$

$$n_0 = 1$$